AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (Currently Amended) In a multi-tier ~~server~~ system that includes a back end server at a first tier and one or more additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the one or more additional servers before the data objects can be used by the ~~and~~ one or more middle tier servers, ~~that obtain data from a database of the back end server,~~ a method for deploying one or more data types from the back end server to the one or more middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising:

an act of creating a special table in ~~the~~ a database of the back end server, the special table including one or more fields for storing data identifying data types and corresponding code for enabling use of each of the data types, and the back end server acting as a single and centralized source from which all middle tier servers obtain data types and the corresponding code required to enable use of the data types by the one or more middle tier servers;

an act of identifying a data type to be deployed from the back end server to the one or more middle tier servers;

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including ~~the~~ data obtained from the special table, including data identifying the data type and the code for enabling use of the data type; and

an act of transmitting the extended assembly to the one or more middle tier servers in the multi-tier system such that the data type as transmitted to and received by the one or more middle tier servers in the multi tier system is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system.

2.   (Original) A method as recited in claim 1, further including an act of creating logic modules in the one or more middle tier servers that enable utilization of the extended as assembly.

3.   (Original) A method as recited in claim 1, wherein the back end server includes a relational database.

4.   (Currently Amended) A method as recited in claim 3, wherein the back end server comprises an ~~Microsoft~~ SQL ~~Server~~ server.

5.   (Original) A method as recited in claim 1, wherein the one or more middle tier servers includes an email server.

6.   (Currently Amended) A method as recited in claim 5, wherein the email server is an ~~Microsoft E~~email exchange server.

7.   (Original) A method as recited in claim 1, wherein the act of identifying the data type to be deployed includes determining that the one or more middle tier servers has requested or does not yet enable use of the data type.

8.   (Original) A method as recited in claim 7, further including an act of adding a new middle tier server to the multi-tier system, and wherein the new middle tier server comprises the one or more middle tier servers that has requested or does not yet enable use of the data type.

9.   (Original) A method as recited in claim 1, further including an act of creating one or more object tables that are linked to the special table and that include additional information defining the data type to be deployed, such that the extended assembly also includes the additional information.

10.   (Currently Amended) In a multi-tier server system that includes a back end server at a first tier and one or more additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the one or more additional servers before the data objects can be used by the and one or more middle tier servers, that obtain data from a database of the back end server, a method for deploying one or more data types from the back end server to the one or more middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising:

an act of modifying a special table in the a database of the back end server, the special table including one or more fields for storing data that identifies data types and includes corresponding code for enabling use of each of the data types, and the back end server acting as a single and centralized source from which all middle tier servers obtain data types and the corresponding code required to enable use of the data types by the one or more middle tier servers, the act of modifying including at least one of modifying the stored data within the one or more fields and adding new stored data to the one or more fields;

an act of identifying a data type to be deployed from the back end server to the one or more middle tier servers;

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including at least one of the modified stored data and the new stored data as obtained from the special table, including data identifying the data type and the code for enabling use of the data type; and

an act of transmitting the extended assembly to the one or more middle tier servers in the multi-tier system such that the data type as transmitted to and received by the one or more middle tier servers in the multi tier system is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system.

11.   (Original) A method as recited in claim 10, further including an act of determining which of one or more middle tier servers should be sent the extended assembly.

12.   (Original) A method as recited in claim 11, wherein the extended assembly enables use of the data type to be deployed at the one or more middle tier servers that have been determined to be sent the extended assembly.

13.   (Original) A method as recited in claim 10, wherein the back end server includes a relational database.

14.   (Currently Amended) A method as recited in claim 13, wherein the back end server comprises an ~~Microsoft~~ SQL ~~Server~~ server.

15.   (Original) A method as recited in claim 13, wherein the one or more middle tier servers includes an email server.

16.   (Original) A method as recited in claim 10, wherein the act of modifying includes adding new stored data corresponding to a new data type not previously enabled in the multi-tier system prior to adding the new stored data.

17.   (Currently Amended) In a multi-tier server system that includes a back end server at a first tier and one or more additional servers at a mittle tear, each additional server using multiple types of data objects that must be defined on the one or more additional servers before the data objects can be used by the ~~and~~ one or more middle tier servers, ~~that obtain data from a database of the back end server,~~ a method for deploying one or more data types from the back end server to the one or more middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising:

an act of adding a new middle tier server to the multi-tier system, the new middle tier server being configured to utilize extended assemblies that are obtained from the back end server which acts as a single and centralized source from which all middle tier servers obtain data types and corresponding code required to enable use of the data types by the one or more middle tier servers, the extended assemblies being configured to enable the use of one or more data types that are defined by data and enabled by code that is contained in the extended assemblies[[.]];

an act of determining which of the one or more data types are to be deployed from the back end server to the new middle tier server;

an act of obtaining one or more extended assemblies corresponding to the one or more data types that have been determined to be deployed, each of the one or more extended assemblies including data and code obtained from a special table stored in ~~the~~ a database of the back end server, the special table including one or more fields for storing data identifying data types and corresponding code for enabling use of each of the data types; and

an act of transmitting, to the new middle tier server, the one or more extended assemblies that correspond to the one or more data types that have been determined to be deployed such that the one or more data types as transmitted to and received by the new middle tier server are consistent and compatible with one or more data types of the same kind on other middle tier servers in the system and which were received by the other middle tier servers from the back end server.

18.   (Original) A method as recited in claim 17, wherein the act of determining is based at least in part on the capabilities of the new middle tier server.

19.   (Original) A method as recited in claim 17, wherein the act of determining is based at least in part on a request by the new middle tier servers for data to enable use of one or more data types.

20.   (Cancelled).

21.   (Currently Amended) In a multi-tier server system that includes a back end server at a first tier and one or more additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the one or more additional servers before the data objects can be used by the and-one or more middle tier servers-that-obtain-data-from-a database-of-the-back-end-server, a method for deploying one or more data types from the back end server to the one or more middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising:

an act of creating a special table in the-a database of the back end server, the special table including one or more fields for storing data identifying a data type and corresponding code for enabling use of the data type, and the back end server acting as a single and centralized source from which all middle tier servers obtain data types and the corresponding code required to enable use of the data types by the one or more middle tier servers;

a step for deploying the data type from the back end server to the one or more middle tier servers such that the data type as transmitted to and received by the one or more middle tier servers in the multi-tier server system is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system.

22.   (Original) A method as recited in claim 21, wherein the step for deploying comprises corresponding acts that include:

an act of identifying the data type to be deployed;

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including the data from the special table identifying the data type and the code for enabling use of the data type; and

an act of transmitting the extended assembly to one or more middle tier servers in the multi-tier system.

23.   (Original) A method as recited in claim 22, further including an act of creating logic in the one or more middle tier servers that enables utilization of the extended assembly.

24.   (Original) A method as recited in claim 22, further including an act of creating at least one object table that includes at least some information defining the data type, and wherein the extended assembly includes the at least some information.

25. (Currently Amended) A computer program product for use in a multi-tier server system that includes a back end server at a first tier and one or more additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the one or more additional servers before the data objects can be used by the and one or more middle tier servers that obtain data from a database of the back-end server, the computer program product including one or more computer-readable media having computer-executable instructions for implementing a method for deploying one or more data types from the back end server to the one or more middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising:

an act of creating a special table in the a database of the back end server, the special table including one or more fields for storing data identifying data types and corresponding code for enabling use of each of the data types, and the back end server acting as a single and centralized source from which all middle tier servers obtain data types and the corresponding code required to enable use of the data types by the one or more middle tier servers;

an act of identifying a data type to be deployed from the back end server to one or more middle tier servers;

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including the data obtained from the special table, including data identifying the data type and the code for enabling use of the data type; and

an act of transmitting the extended assembly to the one or more middle tier servers in the multi-tier system such that the data type as transmitted to and received by the one or more middle tier servers in the multi tier system is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system.

26. (Original) A computer program product as recited in claim 25, wherein the method further includes an act of creating logic modules in the one or more middle tier servers that enable utilization of the extended assembly.

27. (Currently Amended) A computer program product as recited in claim 25, wherein the back end server includes an sequel SQL server.

28.   (Original) A computer program product as recited in claim 25, wherein the one or more middle tier servers includes an email server.

29.   (Currently Amended) A computer program product as recited in claim 28, wherein the email server is an email ~~Microsoft E~~exchange server.

30.   (Original) A computer program product as recited in claim 25, wherein the act of identifying the data type to be deployed includes determining that the one or more middle tier servers has requested or does not yet enable use of the data type.

31.   (Original) A computer program product as recited in claim 25, wherein the method further includes an act of adding a new middle tier server to the multi-tier system, and wherein the new middle tier server comprises the one or more middle tier servers that has requested or does not yet enable use of the data type.

32.   (Original) A computer program product as recited in claim 25, wherein the method further includes an act of creating one or more object tables that are linked to the special table and that include additional information defining the data type to be deployed, and wherein the extended assembly also includes the additional information.

33.   (Currently Amended) A computer program product as recited in claim ~~25~~32, wherein the method further includes modifying at least one of the special table and the one or more object tables.

34.   (New) A method as recited in claim 1, wherein the extended assembly is a single data structure that includes all the data required to enable the one or more middle tier servers to use the data type.